MODELING, IDENTIFICATION AND CONTROL, 1985, VOL. 6, NO. 1, 39–56
doi:10.4173/mic.1985.1.3

A solution to the blow-up problem in adaptive controllers

STEINAR SAELID†, OLAV EGELAND‡ and BJARNE FOSS§

Keywords: *adaptive control, blow-up problems, variable forgetting factor.*

This paper addresses the blow-up problem associated with the parameter estimation part of an adaptive controller. A partial solution to the problem has been devised by the introduction of a variable forgetting factor. However, this does not eliminate the blow-up possibility. This is shown by simulation experiments on two different models.

A new method which completely solves the blow-up problem is presented. This method is presented in two essentially equivalent formulations. One of the formulations is based upon a vector variable forgetting factor, while the other is a Kalman filter approach. This new method and the scalar variable forgetting factor are compared by means of simulation experiments.

1. Introduction

A serious problem which can arise when using an adaptive controller, is the blow-up of the convariance matrix associated with the parameter estimation part of an adaptive controller. The reason for this unpleasant fact is that every adaptive controller has a mechanism for forgetting old information. One way of doing this is by introduction of a constant forgetting factor (Aström *et al.* 1977). This may result in an exponentially increasing parameter covariance matrix during periods of little or no excitations. If this occurs, the system will be extremely sensitive to the onset of disturbances, and control system instability may be the result.

To avoid this problem, Goodwin, Elliot and Teoh (1983) reset the covariance matrix at regular intervals. This method causes loss of information at the resetting samples. Laudau and Lozano (1981) use two forgetting factors to keep the trace of the covariance matrix constant, while Hägglund (1983) has a method which makes the covariance matrix converge to a constant matrix.

Fortescue *et al.* (1981), have devised a partial solution to the problem by the introduction of a variable forgetting factor. The problem has also been addressed by Cordero and Mayne (1981). This method is based on the idea of maintaining a constant level of information content in the estimator part of the controller.

It is the experience of the authors, and this is also shown by Cordero and Mayne, that the method of Fortescue *et al.* does not always work well and must be combined with a restarting mechanism or some other mechanism to prevent blow-up in the general case. One of the reasons for the blow-up possibility when using the Fortescue algorithm, is that only a measurement of the *total* information content of the parameter estimates is controlled. One has no control over how this total information is distributed among the various parameters.

Received 19 March 1985.
† Kongsberg Albatross A.S., N-3600 Kongsberg, Norway.
‡ University of Trondheim, The Norwegian Institute of Technology, Division of Engineering Cybernetics, N-7034 Trondheim, Norway.
§ SINTEF, Division of Automatic Control, N-7034 Trondhein-NTH, Norway.

This fact can be particularly unfortunate if the diagonal elements of the estimation covariance matrix are very different. In that case information loss by a factor of 10 or more, related to a given parameter, may be masked out by a relatively small information gain in a normally larger parameter.

Another, and more important reason, is that the information measure used by Fortescue *et al.* is not a good measure of information content, unless the process is nearly deterministic. This is also pointed out by Fortescue *et al.* (1981).

In the next section of this paper, the method of Fortescue is recapitulated and commented upon. It is shown by examples that the algorithm may very well produce a blow-up situation. In the third section a possible modification of the algorithm is presented by introducing a vector of variable forgetting factors, or alternatively a Kalman filter formulation. It is shown that this modification eliminates the possibility of a blow-up, and the performance of the method is illustrated by simulation experiments.

2. The blow-up phenomenon

A commonly used parameter estimation algorithm in adaptive controllers is (Ljung 1981)

$$\hat{\boldsymbol{\theta}}_{k+1} = \hat{\boldsymbol{\theta}}_k + K_{k+1}\varepsilon_{k+1} \tag{1}$$

$$K_{k+1} = P_k \boldsymbol{\psi}_{k+1}[1 + \boldsymbol{\psi}_{k+1}^T P_k \boldsymbol{\psi}_{k+1}]^{-1} \tag{2}$$

$$P_{k+1} = [I - K_{k+1}\boldsymbol{\psi}_{k+1}^T]P_k \frac{1}{\lambda} \tag{3}$$

where $\hat{\boldsymbol{\theta}}_k$ is an estimate at time k of the parameter vector $\boldsymbol{\theta}$, K_k is the updating gain, $\varepsilon_k = y_k - \bar{y}_k$ is the prediction error, where y_k is the actual process measurement, and $\bar{y}_k$ is a measurement prediction produced by a process model. P_k is an estimate of $\text{cov}(\hat{\boldsymbol{\theta}}_k)$, and $\boldsymbol{\psi}_k$ is equal to $-\partial\varepsilon_k/\partial\hat{\boldsymbol{\theta}}_k$. This sensitivity is directly given in terms of old values of control inputs, u_k, and measurements, y_k, when an ARMA-model and a least squares method is used (Ljung 1977; Söderström *et al.* 1978). Generally, $\boldsymbol{\psi}_k$ is computed by solving a set of sensitivity equations (Saelid and Jenssen 1982; Saelid *et al.* 1983). λ is a forgetting factor, usually chosen in the range $\lambda \in [0{\cdot}95\text{–}1{\cdot}0]$.

As mentioned in the introduction, P_k will increase exponentially if no system excitation exists and $\lambda < 1$. A blow-up situation is then created.

Instead of choosing $\lambda < 1$, parameter tracking capability of the algorithm can be ensured by formulating the parameter estimation problem in the form of an extended Kalman filter, where the unknown parameters are modeled as

$$\boldsymbol{\theta}_{k+1} = \boldsymbol{\theta}_k + \boldsymbol{v}_k$$

and where $\boldsymbol{v}_k$ is a white process noise vector. In that case (3) is modified to be given by

$$P_{k+1} = [I - K_{k+1}\boldsymbol{\psi}_{k+1}^T]P_k + V_k \tag{4}$$

where $V_k = \text{cov}(\boldsymbol{v}_k)$. This formulation also suffers from the blow-up possibility. The algorithm is, however, more robust than the forgetting factor method because P_k only increases linearly when $\boldsymbol{\psi} \approx \mathbf{0}$.

Fortescue's method of blow-up control is based on the following definition of the information content of the estimator (Fortescue *et al.* 1981)

$$\Sigma_k = \lambda_k \Sigma_{k-1} + (1 - \psi_k^T k_k)\varepsilon_k^2 \tag{5}$$

where Σ_k is a measure of the total information content of the estimator, and λ_k is a variable forgetting factor. λ_k is chosen so that $\Sigma_k = \Sigma_{k-1} = \ldots = \Sigma_0$. In other words, the amount of forgetting will at each step correspond to the amount of new information introduced by the latest measurement, so that Σ_k is kept constant. Solution of (5) with respect to λ_k, and assuming $\Sigma_k = \Sigma_{k-1} = \ldots = \Sigma_0$ yields

$$\lambda_k = 1 - \frac{(1 - \psi_k^T K_k)}{\Sigma_0} \varepsilon_k^2$$

or by substitution of (2)

$$\lambda_k = 1 - \frac{\varepsilon_k^2}{\Sigma_0(1 + \psi_k^T P_{k-1}\psi_k)} \tag{6}$$

When a system is deterministic, λ_k approaches 1 if the system is not excited. As a result of this P_k is kept constant. However, if the system has noisy measurements, λ_k will be less than 1, even if no new valuable information is entered into the algorithm. This is seen from (6), and is due to the fact that the algorithm's measure of information content given by (5) is a relevant measure of information content only in the deterministic case. In the stochastic case, ε_k^2 will also contain noise components which are not related to any useful information. In the following, we will see that blow-up may occur when applying Fortescue's algorithm to a stochastic system.

Example 1. Adaptive autopilot.

As an example we shall illustrate the blow-up mechanism by simulation of an adaptive autopilot for ships (Saelid and Jenssen 1982). The autopilot is based on a physically meaningful model of the ship and the environment. The model consists of a low-frequency part (LF-part) representing the steering dynamics in calm water, and a high-frequency part (HF-part) representing the yaw motion due to waves.

The autopilot structure is shown in Fig. 1. The autopilot includes a Kalman filter, and the prediction error is used in a parameter estimation algorithm as given by Eqns. (1)–(3). In addition, a set of sensitivity equations has to be solved in order to obtain $\psi_k = \partial\varepsilon_k/\partial\hat{\theta}_k$. For details regarding autopilot design, see Saelid and Jenssen (1982).

A slightly modified version of this adaptive autopilot (Saelid *et al.* 1984) is simulated using the Fortescue algorithm for computation of λ_k.

Five parameters are estimated: Two parameters in the LF-part system matrix, two parameters in the control matrix and the dominating wave frequency of the HF-model.

The following situation is simulated by running the adaptive controller against a non-linear model of the ship and the environment:

First, the simulator is run for 2000 s using some initial manoeuvres and simulated waves having a significant wave-height $H_{1/3}$ of 8 m. The measurement error has a simulated standard deviation of 0·003 rads ($\simeq$0·2°). λ is set to 1. During this period, the parameters and the covariance matrix obtain an approximately constant level. Σ_0 is set to 0·02.

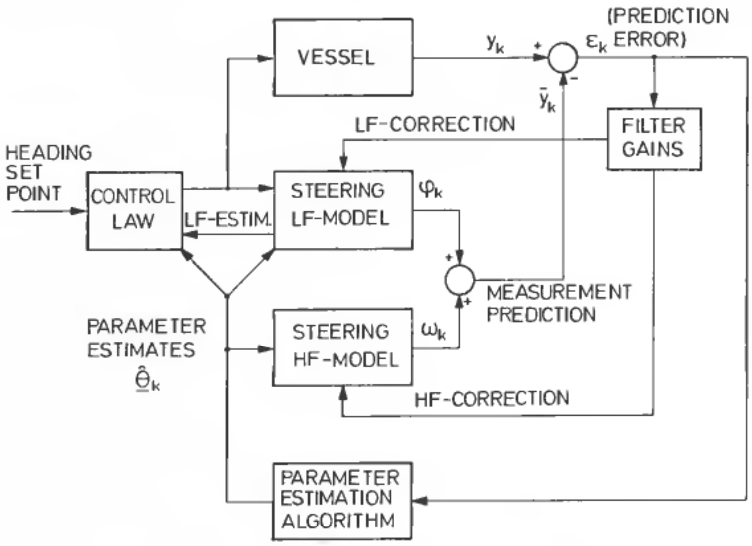


Figure 1. Structure of the adaptive autopilot.

At $t = 2000$ s, the reference excitations are turned off. During the next 3500 s, the covariance matrix grows exponentially as indicated in Fig. 2, where the element $P_{1,1}$ of the P matrix is shown as a function of time. This will happen whenever measurement noise is present. Choosing a larger Σ_0 will not cure the situation. It will only delay the blow-up.

Around $t = 5000$ s, a burst of particularly large waves appears. Due to the blown-up covariance matrix, the parameters change drastically and the system gets unstable after $t = 5500$ s. The parameter estimates and the variable forgetting factor are also shown in Fig. 2. Figure 3 shows the rudder control and the heading of the vessel.

Example 2. Second order process.

As another example we shall examine the simulated application of an adaptive controller to the following process

$$y_k = \frac{b_0 + b_1 z^{-1}}{1 + a_1 z^{-1} + a_2 z^{-2}} u_{k-3} + e_k \tag{7}$$

where y_k is the measurement, u_k is the control, and e_k is the measurement noise at the discrete time k. We assume that e_k is ucorrelated white noise, and that var $(e_k) = 0{\cdot}25$. The true parameter values are $b_0 = 0{\cdot}050$, $b_1 = 0{\cdot}038$, $a_1 = -1{\cdot}38$, and $a_2 = 0{\cdot}47$.

The control law is based on generalized minimum variance control (Clarke and Gawthrop 1979). It minimizes

$$J = E\{[Py_{k+m+1} - w_k + Qu_k]^2\}$$

where m is the process time-delay and w_k the reference signal. P and Q are polynomials chosen so that the desired response characteristics of the adaptive controller are achieved. They are given by $P = 2{\cdot}5 - 1{\cdot}5z^{-1}$ and $Q = 0{\cdot}5(1 - z^{-1})$.

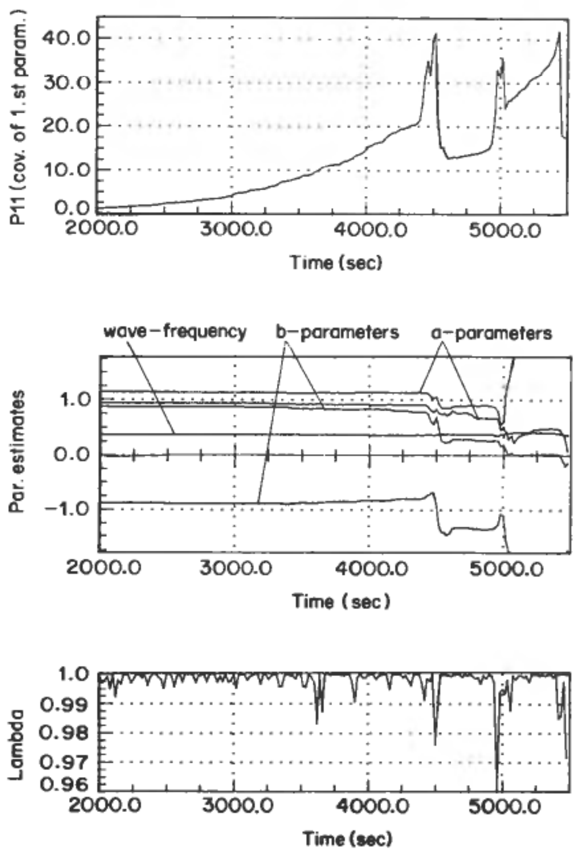


Figure 2. Element p_{11} of P matrix, parameter estimates and forgetting factor λ when the Fortescue method is used.

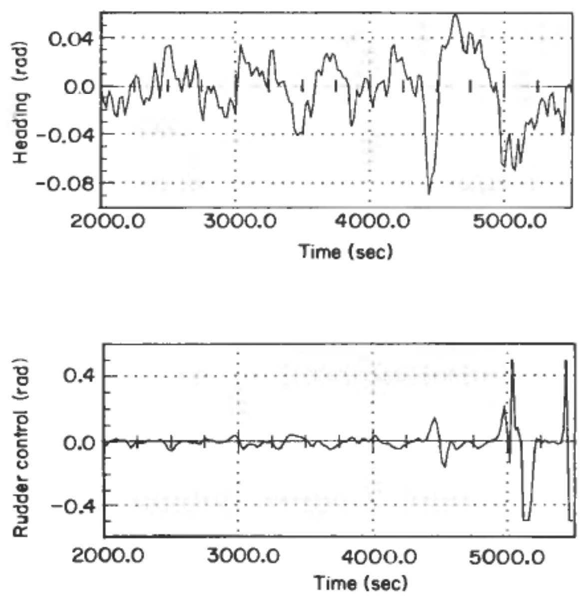


Figure 3. Rudder control and vessel heading.

The adaptive controller is based on explicit model identification. The extended least squares method is therefore applied, together with the Fortescue method, in order to identify the process in Eqn. (7). The model which is used for this purpose has the form

$$\hat{A}(z^{-1})y_k = \hat{B}(z^{-1})u_{k-3} + \hat{C}(z^{-1})\varepsilon_k$$

where

$$\hat{A}(z^{-1}) = 1 + \hat{a}_1 z^{-1} + \hat{a}_2 z^{-2},$$

$$\hat{B}(z^{-1}) = \hat{b}_0 + \hat{b}_1 z^{-1} \quad \text{and}$$

$$\hat{C}(z^{-1}) = 1 + \hat{c}_1 z^{-1} + \hat{c}_2 z^{-2}$$

Hence, six parameters are estimated. The parameter Σ_0 in (6) is set equal to 20.

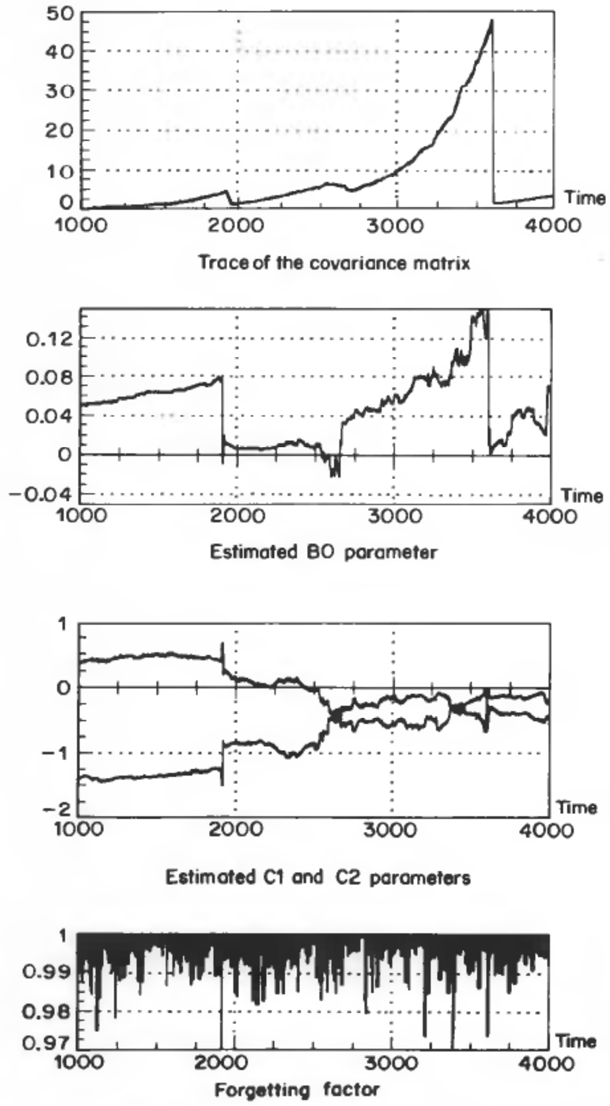


Figure 4. Trace (P), estimates of b_0, c_1 and c_2, and forgetting factor for second order process.

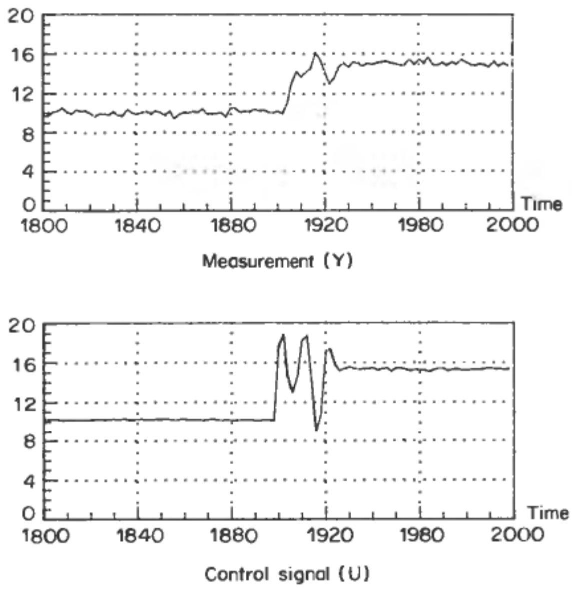


Figure 5. Control and measurement signals.

The system is simulated with initial excitations, such that the parameters have converged to somewhere near their correct values at $t = 1000$. From then on the following reference values are used.

$$w_k = \begin{cases} 10 & t \in [1000, 1900] \\ 15 & t \in [1900, 3600] \\ 10 & t \in [3600, 4000] \end{cases}$$

The resulting trace of the covariance matrix is shown in Fig. 4 together with some of the estimated parameters and the corresponding computed variable forgetting factor. The control signal and the measurement signal around $t = 1900$ are shown in Fig. 5. As it is seen, the algorithm clearly shows blow-up tendencies and the parameter estimator becomes very sensitive.

In the present example we have the selection $C(z^{-1}) = A(z^{-1})$ in the simulation model. If we assume no feedback ($u_k \equiv 0$), ε_k will contain no information about the system parameters, because in that case, e_k can be regarded as pure measurement noise. In an adaptive controller, feedback is present, of course, and the measurement noise will excite the system through the feedback channel. But a large amount of the prediction error, ε_k, will still represent measurement noise and not parameter information. This inevitably leads to blow-up when the Fortescue scheme is applied.

It is clearly seen from an extensive body of simulation experiments, that increased measurement noise increases the blow-up tendency.

Simulation experiments with a lower order $C(z^{-1})$-polynomial than $A(z^{-1})$-polynomial in the process model (7), eliminated the blow-up tendency. This may be explained from the fact that

$$\frac{C(z^{-1})}{A(z^{-1})} = 1 + \frac{C'(z^{-1})}{A(z^{-1})}$$

when the order of the $C(z^{-1})$- and $A(z^{-1})$-polynomials are equal. The constant term, 1, can be interpreted as measurement noise, which is the cause of the blow-up situation shown in this example.

3. The constant information method

It is seen from § 2 that in certain cases there is a need to improve the existing methods for adjusting the forgetting behaviour of an adaptive controller algorithm.

In this section we introduce a new method which is based upon the principle that the amount of forgetting for each parameter is set equal to the amount of information related to that parameter obtained from the last measurement.

An intuitively desirable property of a parameter tracking algorithm will in most cases be to maintain a constant level of parameter updating, only modified by the parameter sensitivity, ψ_k, and the prediction error ε_k. The parameter adjustment is given by

$$\hat{\theta}_{k+1} - \hat{\theta}_k = P_k \psi_{k+1}[1 + \psi_{k+1}^T P_k \psi_{k+1}]^{-1} \varepsilon_{k+1}$$

Hence, from an intuitive and practical point of view, P_k should be kept constant in some sense. In this method we keep the diagonal elements of P_k constant, which is closely related to keeping the variances of the parameters constant.

This is done after an initialization period where the algorithm is run with no forgetting until the parameter estimates are reasonably stabilized. During this period the process has to be sufficiently excited. This can be done by introducing deterministic excitations by cycling the reference signal. After this initial period, the diagonal of P_k is frozen.

The method can be formulated either by using a vector of variable forgetting factors (Saelid and Foss 1983), or by a Kalman filter approach.

What the method actually does, is to disregard (or forget) exactly the same amount of information as is supplied by the measurement at every sample.

3.1. *Vector of variable forgetting factors*

Eqn. (3) with $\lambda = 1$ can be written

$$P_{k+1} = P_k - \frac{P_k \psi_{k+1} \psi_{k+1}^T P_k}{1 + \psi_{k+1}^T P_k \psi_{k+1}} \tag{8}$$

The diagonal elements p_k^{ii} of P_k are given by

$$p_{k+1}^{ii} = p_k^{ii} - (1 + \psi_{k+1}^T P_k \psi_{k+1})(K_k^i)^2 \tag{9}$$

where K_k^i is the element of the K_k matrix.

To obtain $p_{k+1}^{ii} = p_k^{ii}$ we introduce a forgetting factor for each parameter given by $\lambda_k^i = 1 - 1/N_k^i$, where λ_k^i is the forgetting factor for the i'th parameter and N_k^i is the associated memory length.

The covariance updating algorithm (3) now becomes

$$P_{k+1} = L_k^{-1}[I - K_{k+1}\psi_{k+1}^T] P_k L_k^{-1} \tag{10}$$

where $L_k = \mathrm{diag}(\sqrt{\lambda_1}, \ldots, \sqrt{\lambda_p})$ and $p = \dim \theta$.

With this modification, Eqn. (9) is changed to

$$p_{k+1}^{ii} = [p_k^{ii} - (1 + \psi_{k+1}^T P_k \psi_{k+1})(K_k^i)^2]/\lambda_k^i \tag{11}$$

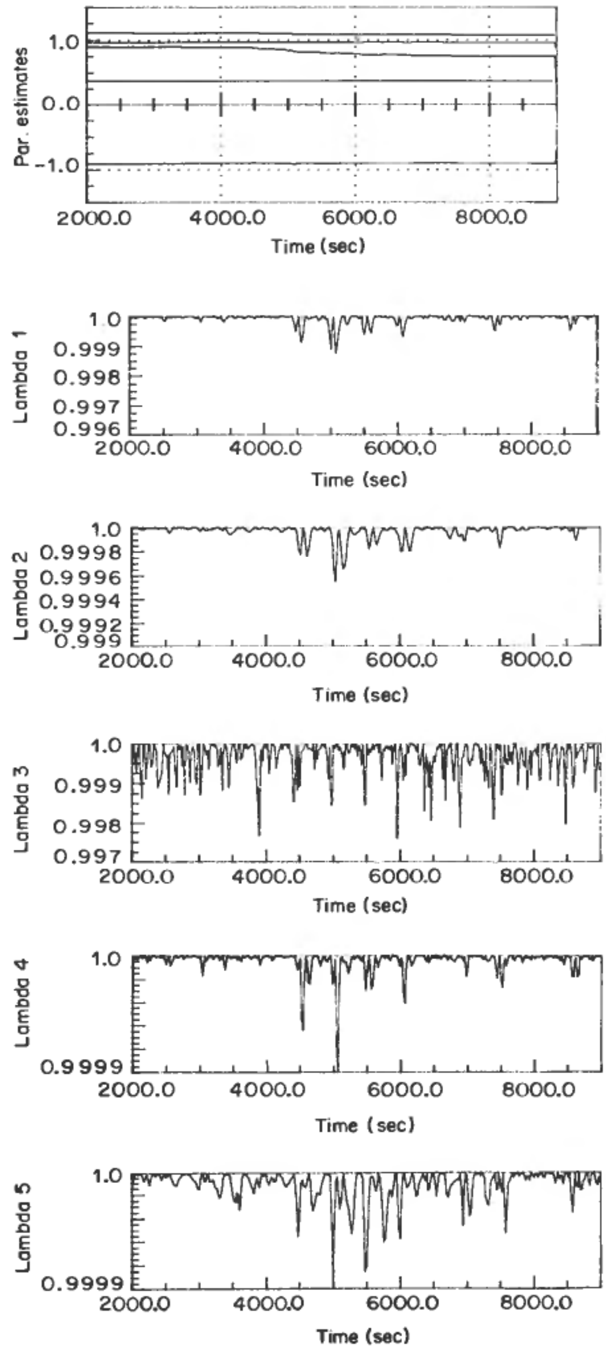


Figure 6. Parameter estimates and forgetting factors for adaptive autopilot.

By choosing

$$\lambda_k^i = 1 - (1 + \psi_{k+1}^T P_k \psi_{k+1})(K_k^i)^2/p_k^{ii} \tag{12}$$

or alternatively

$$\lambda_k^i = p_k^{ii}/p_{k+1}^{ii}(\lambda^i = 1)$$

where $p_{k+1}^{ii}(\lambda^i = 1)$ is the p_{k+1}^{ii} we get, with $\lambda^i = 1$, we get

$$p_{k+1}^{ii} = p_k^{ii}$$

By using (12) in the covariance updating algorithm, the diagonal elements of P_k will remain constant for all k, whereas the off-diagonal element will change according to variations in ψ_k.

The algorithm was tested by running the simulation experiments of § 2 once more, using a vector of variable forgetting factors instead of the Fortescue method. The algorithm behaves very well both in the adaptive autopilot and the second order process.

Figure 6 shows the parameter estimates and the forgetting factors from the adaptive autopilot simulation. Figure 7 shows vessel heading and rudder control.

Figure 8 shows the B and C parameter estimates and three of the forgetting factors for the second order process. Figure 9 shows the control and measurement around the reference change at $t = 1900$ s.

3.2. Kalman filter

As described above, Eqn. (3) can be replaced by Eqn. (4) by modeling $\boldsymbol{\theta}_k$ as a stochastic process.

By choosing

$$V_k = \operatorname{diag}(v_k^{11}, \ldots, v_k^{pp}) \tag{13}$$

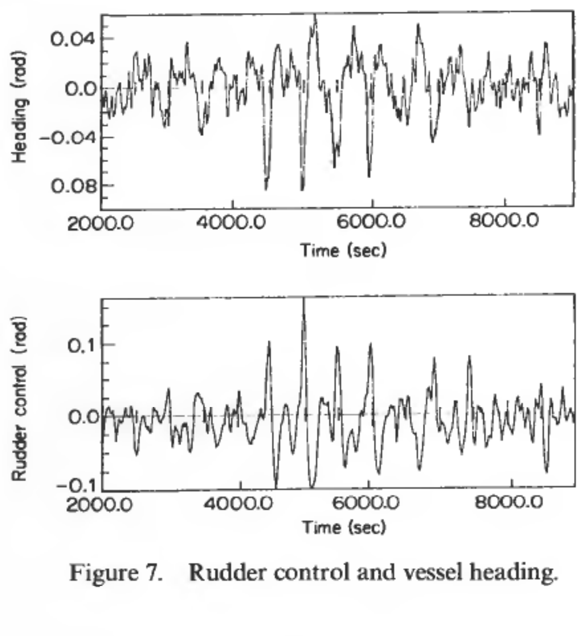


Figure 7. Rudder control and vessel heading.

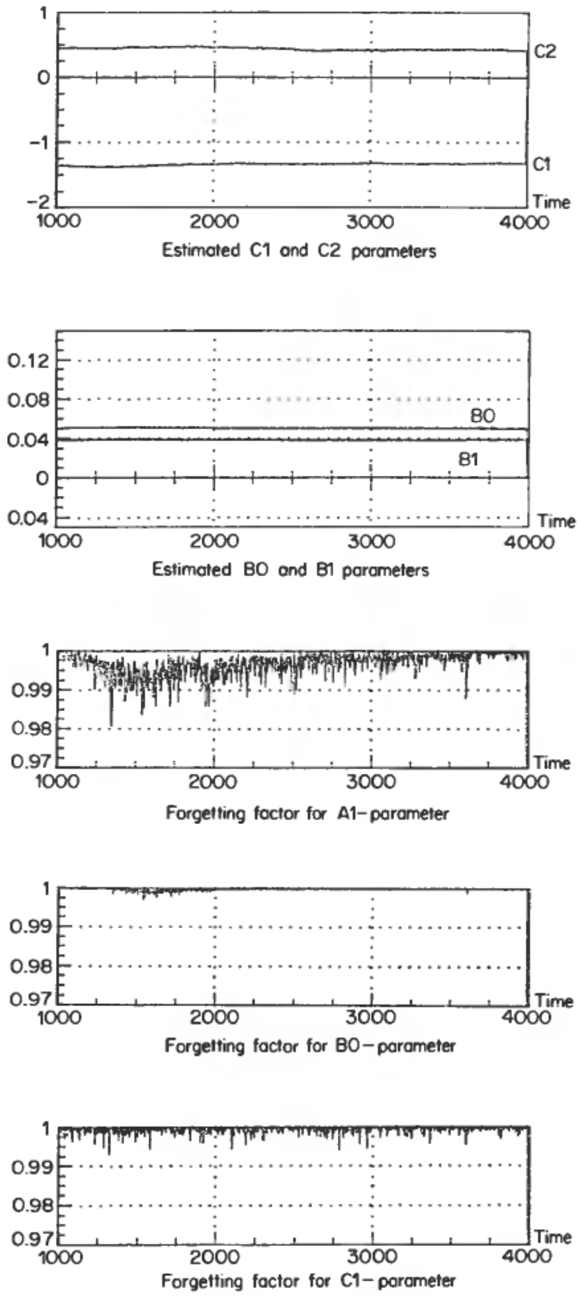


Figure 8. Estimated c_1, c_2, b_0, and b_1 parameters and forgetting factors corresponding to a_1, b_0, and c_1 for second order process.

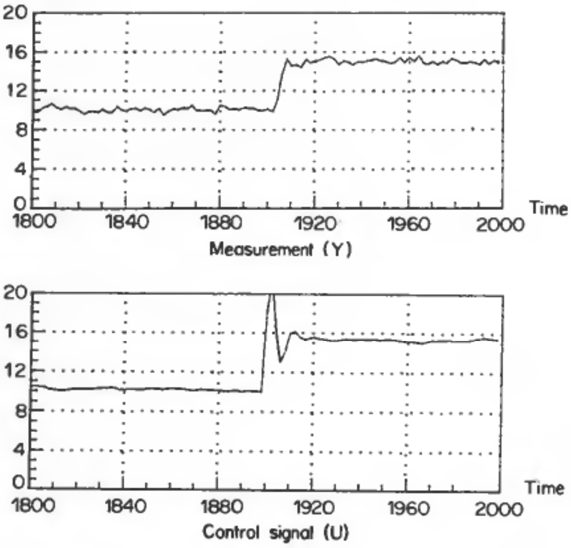


Figure 9. Measurement and control signals.

where $\dim \boldsymbol{\theta} = p$,

$$v_k^{ii} = (K_{k+1}\psi_{k+1}^T P_k)^{ii}$$

and $(K_{k+1}\psi_{k+1}^T P_k)^{ii}$ is the i'th element of the diagonal of $K_{k+1}\psi_{k+1}^T P_k$, we obtain $p_{k+1}^{ii} = p_k^{ii}$.

This gives a very simple algorithm which can also be used for systems with multiple measurements.

Using this approach, the updating of the off-diagonal element of P_k is given by

$$p_{k+1}^{ij} = [(I - K_{k+1}\psi_{k+1}^T)P_k]^{ij}, \qquad i \neq j \tag{14}$$

while the formulation with a vector of variable forgetting factors gives

$$p_{k+1}^{ij} = [(I - K_{k+1}\psi_{k+1}^T)P_k]^{ij}/(\lambda_k^i \lambda_k^j)^{1/2}, \qquad i \neq j \tag{15}$$

The only difference between the two formulations is the $(\lambda_k^i \lambda_k^j)^{-1/2}$ term in (15).

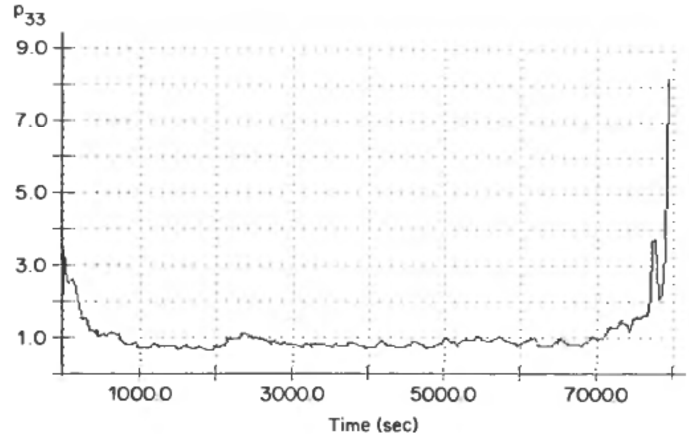
During the initializing period, the forgetting factors in (10) are set to 1, which corresponds to $V_k = 0$ in (13).

Example 3. Step in parameters in the adaptive autopilot.

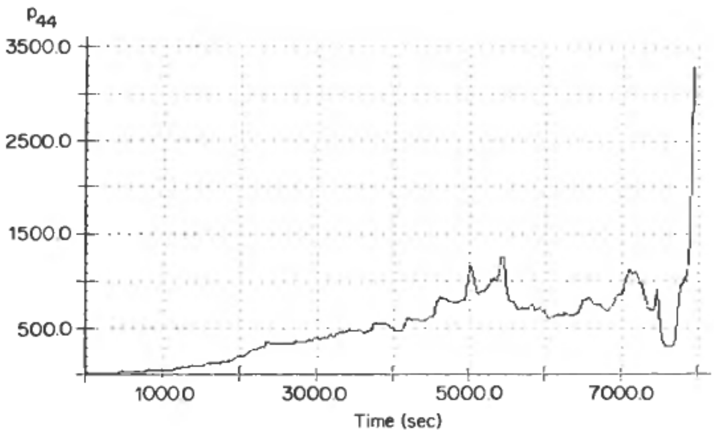
The adaptive autopilot of Example 1 was simulated once more. At the start of the simulation experiment, the parameters had converged. Significant waveheight was 8 m.

Initial speed of the vessel was 15 knots. After 2000 s the speed was halved. The autopilot has an adaptive feedforward (scheduling) from the vessel speed. This feedforward was kept at 15 knots, causing a considerable change in parameter values.

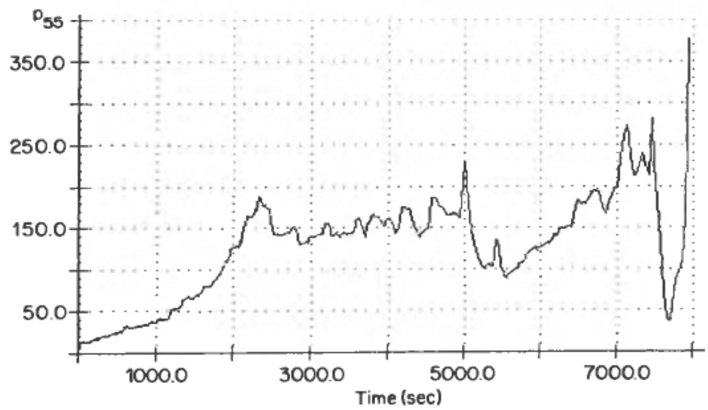
The system was simulated using the Fortescue method. From Fig. 10 we see that while p_{33}, that is the third element of the diagonal of P_k, is well-behaved, p_{44} and



(*a*) p_{33}



(*b*) p_{44}



(*c*) p_{55}

Figure 10. Diagonal elements (*a*) p_{33}, (*b*) p_{44}, and (*c*) p_{55} of P matrix for adaptive autopilot when the Fortescue method is used.

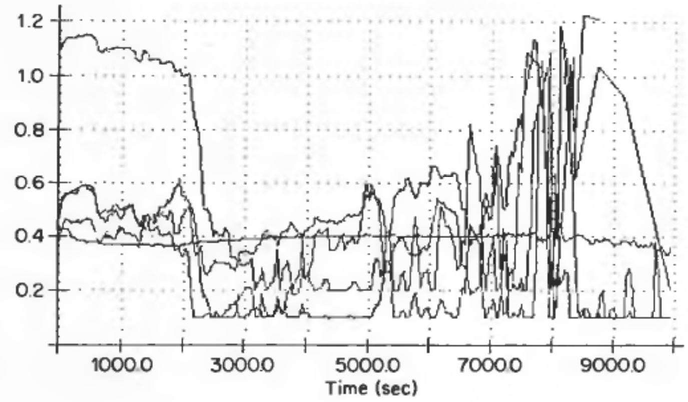
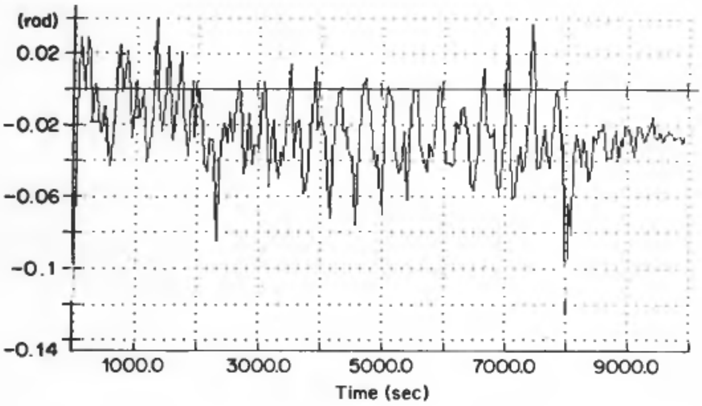
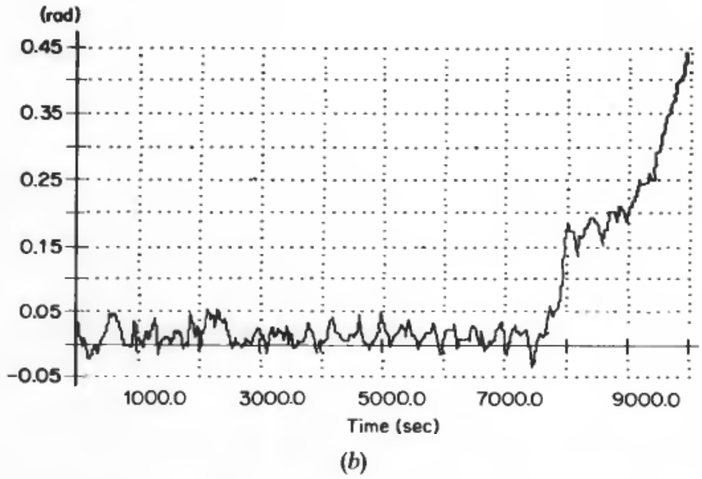


Figure 11. Parameter estimates.



(*a*)



(*b*)

Figure 12. (*a*) Rudder control and (*b*) vessel heading.

p_{55} increase near exponential for $t < 2000$ s, which is the time of the parameter change.

The parameter estimates are shown in Fig. 11, and rudder control and vessel heading are shown in Fig. 12. Due to the blown-up P_k, the parameter updating gains are too high at the time of the change in parameter values. This causes the system to become unstable after 7500 s, which is seen from Fig. 12.

Σ_0 was chosen as 0·02. A large Σ_0 would have decreased the blow-up tendency in P_k, but this gives a slow speed of adaption for some of the parameters.

The system was then simulated, first with a vector of variable forgetting factors, and then with the Kalman filter formulation.

As could be expected, the formulation using a vector of variable forgetting factors and the Kalman filter formulation given almost identical results. Both methods are well-behaved.

The parameter estimates are shown in Fig. 13. These parameter estimates were obtained using the Kalman filter formulation. v_{33}, v_{44} and v_{55} are shown in Fig. 14. Rudder control and vessel heading are shown in Fig. 15. Rudder control and vessel heading were as good as identical for the Kalman filter formulation and the formulation with a vector of variable forgetting factors.

4. Conclusion

It has been demonstrated that the variable forgetting factor algorithm as proposed by Fortescue *et al.* (1981) does not prevent blow-up if measurement noise is present.

A new method that solves the blow-up problem is introduced. The method can be formulated by means of a vector of variable forgetting factors or by means of a Kalman filter. These two formulations are essentially equivalent.

The method is based upon the principle that at every sample exactly the same amount of information is disregarded for each parameter as is supplied by means of the measurement.

The excellent behaviour of the method is exemplified by application of the algorithm to a simulated adaptive autopilot and a simulated second order process.

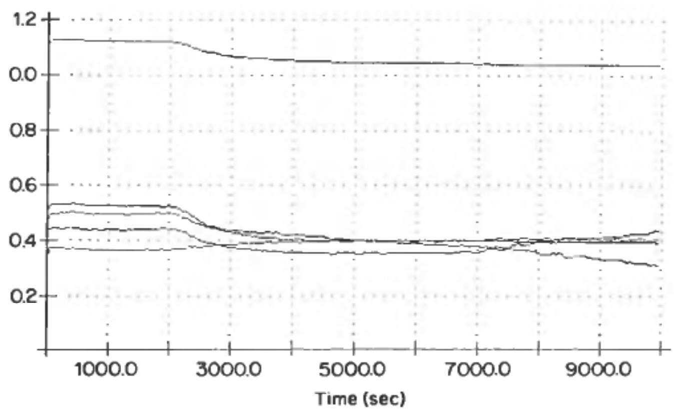


Figure 13. Parameter estimates for adaptive autopilot when Kalman filter formulation is used.

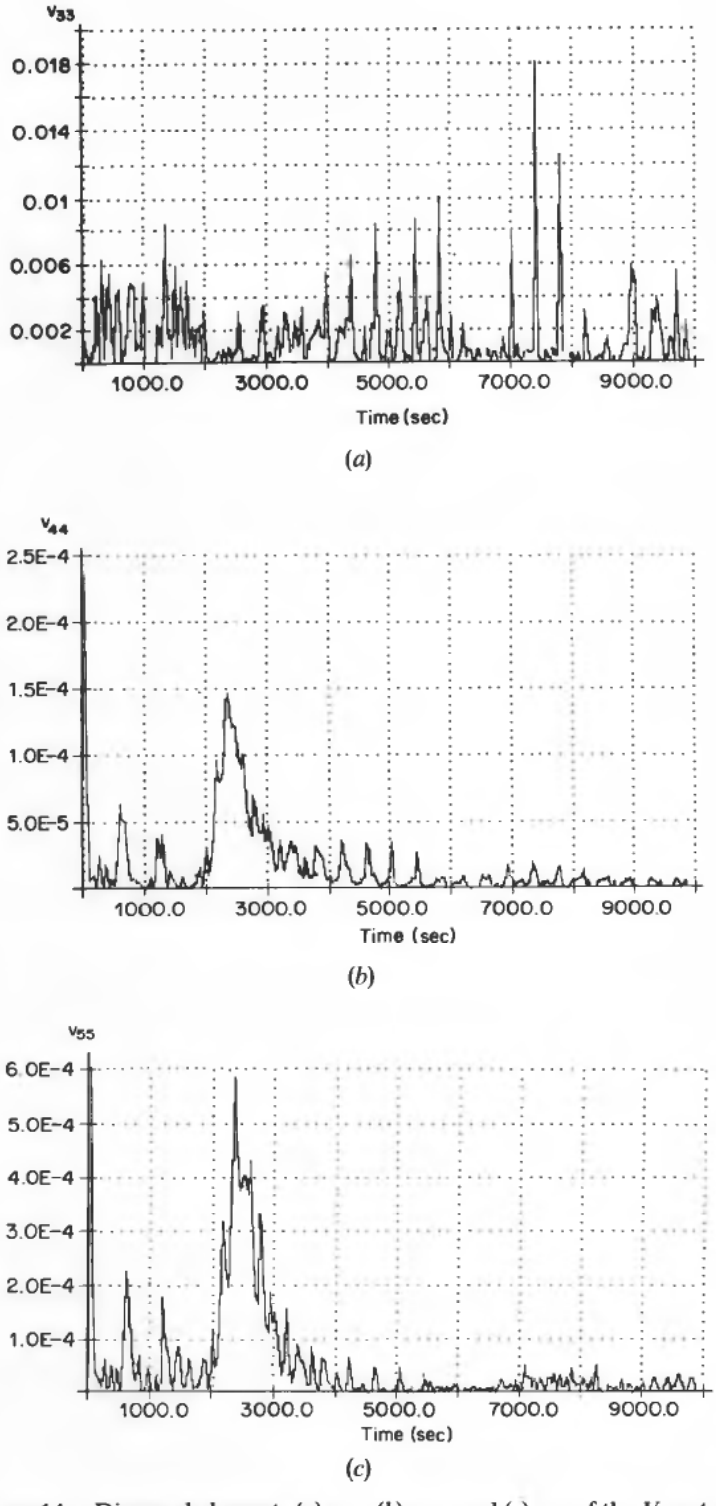
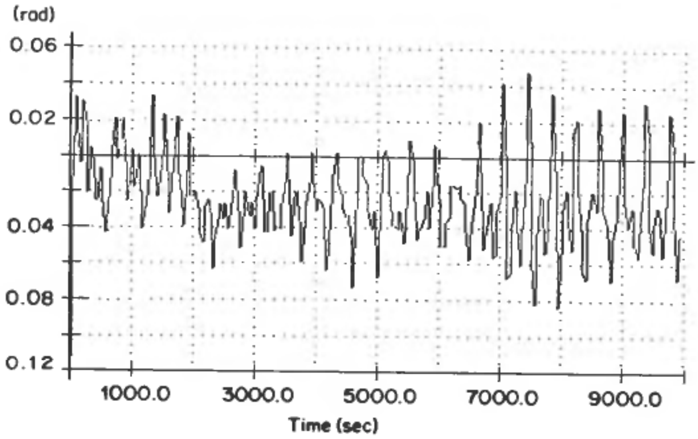
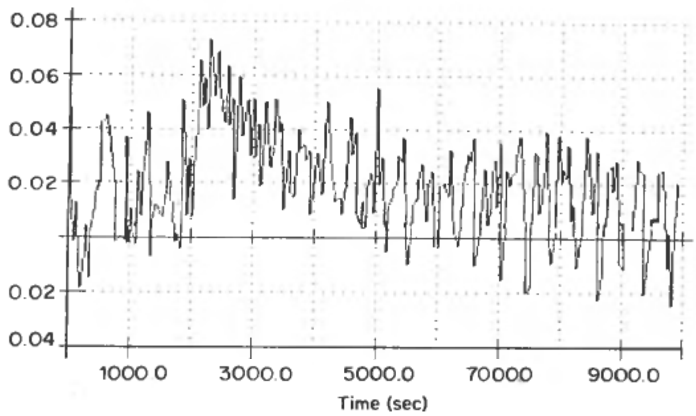


Figure 14. Diagonal elements (*a*) v_{33}, (*b*) v_{44}, and (*c*) v_{55} of the V matrix.



(*a*)



(*b*)

Figure 15. (*a*) Rudder control and (*b*) vessel heading.

REFERENCES

ASTRÖM, K. J., BORRISSON, V., LJUNG, L., and WITTENMARK, B. (1977). Theory and application of self-tuning regulators. *Automatica*, **13**, p. 457.

CLARKE, D., and GAWTHROP, P. (1979). Self-tuning control. *Proc. I.E.E.* (*Part D*), **126**, 663–640.

CORDERO, A. O., and MAYNE, D. Q. (1981). Deterministic convergence of a self-tuning regulator with variable forgetting factor. *Proc. I.E.E.* (*Part D*), **128**, 19–23.

FORTESCUE, T. R., KERSCHENBAUM, L. S., and YDSTIE, B. E. (1981). Implementation of self-tuning regulators with variable forgetting factors. *Automatica*, **17**, 831–835.

GOODWIN, G. C., ELLIOT, H., and TEOH, E. K. (1983). Deterministic convergence of a self tuning regulator with covariance resetting. *Proc. I.E.E.E.* (*Part D*), **130**, 6–8.

HÄGGLUND, T. (1983). Recursive least squares identification with forgetting old data. *Dept. of Automatic Control, Lund Institute of Technology*, March 1983.

LANDAU, I. D., and LOZANO, R. (1981). Unification of discrete time explicit model reference adaptive control design. *Automatica*, **17**, 593–611.

LJUNG, L. (1977). Analysis of recursive stochastic algorithms. *I.E.E.E. Trans.*, **AC-22**, 551–575.

LJUNG, L. (1981). Analysis of a general recursive prediction error identification algorithm. *Automatica*, **17**, 89–99.

SAELID, S., SVANES, T., ONSHUS, T., and JENSSEN, N. A. (1984). Design considerations, analysis and practical experience with an adaptive ships autopilot. *Preprints, IFAC 9th World Congress, Budapest, Hungary*, July 1984, vol. 2, pp. 35–40.

SAELID, S., and JENSSEN, N. A. (1982). Adaptive autopilot with wave filter. *Proc. IFAC Symp. on Identification and System Parameter Estimation, Washington*, pp. 1220–1225.

SAELID, S., JENSSEN, N. A., and BALCHEN, J. G. (1983). Design and analysis of a dynamic positioning system based on Kalman filtering and optimal control. *I.E.E.E. Trans.*, **AC-28**, 331–339.

SAELID, S., and FOSS, B. (1983). Adaptive controllers with a vector variable forgetting factor. *Proc. of the 22nd I.E.E.E. CDC Conf., San Antonio, Texas*, pp. 1488–1494.

SÖDERSTRÖM, T., LJUNG, L., and GUSTAVSSON, I. (1978). A theoretical analysis of recursive identification methods. *Automatica*, **14**, 231–244.